IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: ITIKARLAPALLI *et al*	Art Unit: 2168
Appl. No.: 10/709,522	Examiner: SANDERS, AARON J
Filed: 05/11/2004	Attorney Docket No.: ORCL-003
For: Simplifying Implementation of Custom Atomic Transactions in a Programming Environment	

Appeal Brief Under 37 CFR § 41.37

Mail Stop **Appeal Brief - Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Further to the Notice of Appeal filed on 21-June-2010, Appellants submit this appeal brief under 37 CFR § 41.37.

As required by 37 C.F.R. § 41.37, this brief contains items under the following headings:

I. REAL PARTY IN INTEREST

II. RELATED APPEALS AND INTERFERENCES

III. STATUS OF CLAIMS

IV. STATUS OF AMENDMENTS

V. SUMMARY OF CLAIMED SUBJECT MATTER

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

VII. THE ARGUMENT

VIII. CLAIMS

IX. EVIDENCE APPENDIX

X. RELATED PROCEEDINGS APPENDIX

I. REAL PARTY IN INTEREST

This application was assigned to ORACLE INTERNATION CORPORATION by virtue of the assignment recorded on 11-May-2004 at Reel/Frame: 014596/0076 at the USPTO. ORACLE INTERNATIONAL CORPORATION is accordingly the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN THE APPLICATION

There are 18 claims currently pending in the application.

B. STATUS OF ALL CLAIMS

Claims pending:	1-10, 13-21 and 25
Claims canceled:	11-12 and 22-24
Claims withdrawn:	None
Claims allowed:	None
Claims objected to:	None
Claims rejected:	1-10, 13-21 and 25

C. CLAIMS ON APPEAL

All the rejected claims 1-10, 13-21 and 25 are subject of this appeal.

IV. STATUS OF AMENDMENTS

The Appellants have not filed any amendments after the Final Office Action dated 04/19/2010 (hereafter “Outstanding Final Office Action”).

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention is directed to facilitating programmers to simplify implementation of custom atomic transactions. The invention can be appreciated in comparison with a prior approach depicted in Figure 1 of the subject patent application.

As may be readily appreciated from there, the programmer is inserting various programming statements to keep track of execution flow. For example, when execution flow is past line 153, the variable temp-was-one would equal 0 if lines 143-152 were not executed and would equal 1 otherwise.

Then, based on such tracking of execution flow, the programmer is shown executing desired custom roll-back procedures for the two flows (i.e., corresponding to the two values for the variable temp-was-one). In particular, three custom roll-back procedures are shown executed in addition for the case temp-was-one equals 1.

It should be appreciated that the programmer has the mental burden of keeping track of the order of execution of the task procedures such that the corresponding roll-back procedures are executed in the reverse order (as can be see from lines 165-190 of Figure 1 of the subject application).

From the above, it is pointed out that the programming complexity increases rapidly as the number of procedures and conditional execution flows increases.

However, it is noted that the technique of Figure 1 provides the programmer the ability to have custom roll back procedures (i.e., the procedures with 'reverse' embedded in name in lines 165-190).

Various features of the present invention provide such an ability for a programmer to specify custom roll back procedures, while shielding the programmer from the detail of having to keep track of execution flows, etc. The corresponding techniques are recited in each of the claims explained below.

Independent claim 1 recites a method of implementing atomic transactions (paragraph 0004 of the specification) in a system (700 of Figure 7 and paragraph 0082 of the specification). The method is recited to contain:

requesting (line 209 of Figure 2 and step 510 of Figure 5, and paragraphs 0036, 0039 and 0068of the specification) in a user program (Figure 2) a transaction identifier (line 209 of Figure 2 and column 402 of Figure 4) for an atomic transaction;

generating (610 of Figure 6 and paragraph 0075) said transaction identifier in a transaction manager (Figure 3 and paragraph 0052) in response to said requesting;

specifying (step 540 of Figure 5 and paragraph 0069) in said user program a plurality of combinations (lines 215, 225, 255, 260, 265 and 270 of Figure 2 and paragraph 0036), wherein each of said plurality of combinations contains said transaction identifier (stored in variable “txid” of Figure 2), a task procedure (“P1()”, “P2()”, “P3()”, “P4()”, “P5()”, and “P6()” of Figure 2) and a rollback procedure (“R1()”, “R2()”, “R3()”, “R4()”, “R5()”, and “R6()” of Figure 2),

wherein each combination indicates that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted (paragraph 0006 of the specification),

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure (paragraph 0040 of the specification);

executing a set of task procedures in a sequential order (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 403 of Figure 4, paragraph 0059-0064 of the specification) according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations (steps 650, 660 and 690 of Figure 6, execution interface 360 of Figure 3, and paragraphs 0053, and 0076 to 0079 of the specification);

keeping track of a set of rollback procedures corresponding to said set of task procedures (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 404 and stack status 405 of Figure 4, paragraph 0059-0064 of the specification), each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program, wherein said set of rollback procedures are kept track of external (table 400 of Figure 4 maintained by transaction manager of Figure 3 and paragraphs 0019, 0058 and 0059 of the specification) to said user program (Figure 2 and paragraph 0035 of the specification) in response to said executing of the corresponding task procedures; and

executing said set of rollback procedures in a reverse order (stack status 405 of Figure 4 and paragraph 0064 of the specification) of said sequential order if said atomic transaction is to be aborted (step 670 of Figure 6 and 0078 of the specification),

wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program (lines 215, 225, 255, 260, 265 and 270 of Figure 2),

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure (paragraph 0006 of the specification), and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures (paragraph 0020 of the specification).

Accordingly, the programmer is relieved of the challenge of keeping track of the sequence of task procedures executed (based on execution flow), and yet be able to provide the custom roll back procedures associated with respective task procedures. This combination simplifies flexible implementation of custom atomic transactions for programmers.

Independent claim 7 recites a computer readable storage medium (secondary memory 730 of Figure 7 and paragraphs 0086 and 0087 of the specification) carrying one or more sequences of instructions representing a user program (Figure 2) for execution on a system (700 of Figure 7 and paragraph 0082 of the specification), said user program implementing an atomic transaction (paragraph 0004 of the specification). Claim 7 further recites that the execution of said one or more sequences of instructions by one or more processors (processing unit 710 of Figure 7 and paragraph 0084 of the specification) contained in said system causes said system to perform the actions of:

requesting (line 209 of Figure 2 and step 510 of Figure 5, and paragraphs 0036, 0039 and 0068 of the specification) an identifier (line 209 of Figure 2 and column 402 of

Figure 4) in said user program from a transaction manager (Figure 3 and paragraph 0052) for said atomic transaction, said transaction manager being provided external to said user program (paragraph 0082 of the specification), wherein said transaction manager generates a unique value (column 402 of Figure 4) as said identifier and provides said identifier to said user program (program interface block 310 of Figure 3 and paragraph 0051 of the specification);

setting a variable ("txid" in Figure 2) to equal said identifier in said user program (line 209 of Figure 2 and step 530 of Figure 5, and paragraphs 0036, 0039 and 0068 of the specification);

specifying (step 540 of Figure 5 and paragraph 0069) a plurality of combinations (lines 215, 225, 255, 260, 265 and 270 of Figure 2 and paragraph 0036) in said user program, wherein each of said plurality of combinations contains said variable ("txid"), a task procedure ("P1()", "P2()", "P3()", "P4()", "P5()", and "P6()" of Figure 2) and a rollback procedure ("R1()", "R2()", "R3()", "R4()", "R5()", and "R6()" of Figure 2),

wherein each combination indicates in said user program that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted (paragraph 0006 of the specification),

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure (paragraph 0040 of the specification), wherein said variable in each of said plurality of combinations specifies said identifier generated by said transaction manager;

executing a set of task procedures in a sequential order (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 403 of Figure 4, paragraph 0059-0064 of the specification), wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations (steps 650, 660 and 690 of Figure 6, execution interface 360 of Figure 3, and paragraphs 0053, and 0076 to 0079 of the specification); and

aborting said atomic transaction by specifying, in said user program, said identifier associated with an abort procedure (line 280 of Figure 2, steps 560 and 570 of Figure 5 and paragraphs 0037, 0046, 0070 and 0071) to cause a set of rollback procedures to be executed in a reverse order of said sequential order (step 670 of Figure 6 and 0078 of the specification), each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 404 and stack status 405 of Figure 4, paragraph 0059-0064 of the specification),

wherein said plurality of combinations and said abort procedure are contained in said user program (Figure 2),

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure (paragraph 0006 of the specification), and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures (paragraph 0020 of the specification).

Independent claim 10 recites a computer readable storage medium (secondary memory 730 of Figure 7 and paragraphs 0086 and 0087 of the specification) carrying one or more sequences of instructions for supporting implementation of a transaction manager (Figure 3 and paragraph 0052) which supports an atomic transaction (paragraph 0004 of the specification) in a user program (Figure 2) executing in a system (700 of Figure 7 and paragraph 0082 of the specification). Claim 10 further recites that the execution of said one or more sequences of instructions by one or more processors (processing unit 710 of Figure 7 and paragraph 0084 of the specification) contained in said system causes said system to perform the actions of:

generating (ID allocation block 340 of Figure 3, 610 of Figure 6 and paragraphs 0052 and 0075 of the specification) an identifier (column 402 of Figure 4) for said atomic transaction for said user program;

providing said identifier to said user program (line 209 of Figure 2, program interface block 310 of Figure 3 and paragraph 0051 of the specification);

receiving (program interface block 310 of Figure 3, step 630 of Figure 6 and paragraphs 0051 and 0075 of the specification) a plurality of combinations (lines 215, 225, 255, 260, 265 and 270 of Figure 2 and paragraph 0036) for execution from said user program, wherein each of said plurality of combinations contains said transaction identifier (stored in variable “txid” of Figure 2), a task procedure (“P1()”, “P2()”, “P3()”, “P4()”, “P5()”, and “P6()” of Figure 2) and a rollback procedure (“R1()”, “R2()”, “R3()”, “R4()”, “R5()”, and “R6()” of Figure 2), wherein each combination indicates that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted (paragraph 0006 of the specification),

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure (paragraph 0040 of the specification);

executing a set of task procedures in a sequential order (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 403 of Figure 4, paragraph 0059-0064 of the specification) according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations (steps 650, 660 and 690 of Figure 6, execution interface 360 of Figure 3, and paragraphs 0053, and 0076 to 0079 of the specification);

keeping track of a set of rollback procedures corresponding to said set of task procedures (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 404 and stack status 405 of Figure 4, paragraph 0059-0064 of the specification), each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program (table 400 of Figure 4 maintained by transaction manager of Figure 3, transaction management block 330 of Figure 3 and paragraphs 0019, 0054, 0058 and 0059 of the specification); and

executing (transaction management block 330 of Figure 3, step 670 of Figure 6 and paragraphs 0055 and 0078 of the specification) said set of rollback procedures in a reverse order of said sequential order (stack status 405 of Figure 4 and paragraph 0064 of the specification) in response to receiving an abort request (step 660 of Figure 6 and 0077 of the specification), said abort request being received from said user program and containing said identifier (line 280 of Figure 2, paragraph 0046 of the specification),

wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program (lines 215, 225, 255, 260, 265 and 270 of Figure 2),

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure (paragraph 0006 of the specification), and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures (paragraph 0020 of the specification),

wherein said transaction manager is provided external to user programs including said user program (paragraph 0082 of the specification).

Independent claim 16 recites a computer system (700 of Figure 7 and paragraph 0082 of the specification) comprising a memory (RAM 720 of Figure 7 and paragraph 0084 of the specification) storing a plurality of instructions (Figure 2), a processing unit (710 of Figure 7 and paragraph 0084 of the specification) coupled to said memory and executing said plurality of instructions and a computer readable medium (secondary memory 730 of Figure 7 and paragraphs 0086 and 0087 of the specification) to store and provide said plurality of instructions to said memory. Claim 16 further recites that the execution of said plurality of instructions by said processing unit causes said computer system to support implementation of atomic transactions in a programming environment by performing the actions of:

request (line 209 of Figure 2 and step 510 of Figure 5, and paragraphs 0036, 0039 and 0068) and in a user program (Figure 2) a transaction identifier (line 209 of Figure 2 and column 402 of Figure 4) for an atomic transaction;

generate (610 of Figure 6 and paragraph 0075) said transaction identifier in a transaction manager (Figure 3 and paragraph 0052) in response to said requesting, wherein said transaction manager is provided external to said user program (paragraph 0082 of the specification);

specify (step 540 of Figure 5 and paragraph 0069) in said user program a plurality of combinations (lines 215, 225, 255, 260, 265 and 270 of Figure 2 and paragraph 0036), wherein each of said plurality of combinations contains said transaction identifier (stored in variable “txid” of Figure 2), a task procedure (“P1()”, “P2()”, “P3()”, “P4()”, “P5()”, and “P6()” of Figure 2) and a rollback procedure (“R1()”, “R2()”, “R3()”, “R4()”, “R5()”, and “R6()” of Figure 2),

wherein each combination indicates in said user program that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted (paragraph 0006 of the specification),

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said rollback procedure is specified as a separate procedure from said task procedure (paragraph 0040 of the specification);

execute a set of task procedures in a sequential order (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 403 of Figure 4, paragraph 0059-0064 of the specification) according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations (steps 650, 660 and 690 of Figure 6, execution interface 360 of Figure 3, and paragraphs 0053, and 0076 to 0079 of the specification);

keep track of a set of rollback procedures corresponding to said set of task procedures (rows 411, 412 and 418 for a transaction with identifier 200 and rows 413-417 and 419 for a transaction with identifier 300 in column 404 and stack status 405 of Figure 4, paragraph 0059-0064 of the specification), each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program, wherein said set of rollback procedures are kept track of external (table 400 of Figure 4 maintained by transaction manager of Figure 3 and paragraphs 0019, 0058 and 0059 of the specification) to said user program (Figure 2 and paragraph 0035 of the specification) in response to said executing of the corresponding task procedures; and

execute said set of rollback procedures in a reverse order (stack status 405 of Figure 4 and paragraph 0064 of the specification) of said sequential order if said atomic transaction is to be aborted (step 670 of Figure 6 and 0078 of the specification), , wherein said rollback procedures are identified according to said keeping,

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure procedure (paragraph 0006 of the specification), and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures (paragraph 0020 of the specification).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

(A) Rejection of claims 1-2, 5-10, 13, 16-17, 20-21 and 25 under 35 U.S.C. § 103(a) allegedly as being unpatcntablc over Applicant's Admitted Prior Art, Fig. 1 and Specification paragraphs. 3-7 and 22-33 ("AAPA"), in view of Gostanian *et al*, U.S. 5,781,910 ("Gostanian"), and in view of Lordi *et al*, U.S. 5,857,204 ("Lordi").

(B) Rejection of claims 3-4, 14-15 and 18-19 under 35 U.S.C. § 103(a) as being unpatentable over AAPA in view of Gostanian, Lordi, and Raz, U.S. 5,701,480 ("Raz").

VII. THE ARGUMENT

A. Rejection of Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 Under 35 U.S.C. § 103(a)

Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over AAPA in view of Gostanian and Lordi. The Honorable Board is respectfully requested to reverse the rejection on several grounds noted below.

VII.A.1. None of the References Teaches Several Claimed Features

As stated in MPEP 2143, the Patent Office has the burden of showing that the prior art reference (or references when combined) must teach or suggest all the claim limitations. That burden is not adequately met by the Examiner.

Independent claim 1 recites that, “requesting in a user program a transaction identifier for an atomic transaction; generating said transaction identifier in a transaction manager in response to said requesting; specifying in said user program … said transaction identifier, …”.

Similarly, independent claim 7 recites, “requesting an identifier in said user program from a transaction manager for said atomic transaction, … wherein said transaction manager … provides said identifier to said user program; setting a variable to equal said identifier in said user program; specifying a plurality of combinations in said user program for execution in said system, wherein each of said plurality of combinations contains said variable,…”

In other words, independent claims 1 and 7 require that the user program request a transaction identifier, which is generated in a transaction manager and provided to the user program for specifying in the different claimed combinations in the user program.

The Examiner equates the claimed identifier to 'Account1()' in line 105 of Figure 1 the subject patent application. See Page 3 lines 10-11 of the Outstanding Final Office Action.

The Honorable Board's attention is directed to the fact that 'Account1()' is part of the user program and thus **not** generated in (or provided from) transaction manager. At least for such a reason, it is asserted that the Examiner's rejection of claims 1 and 7 is prima facie defective. Independent claim 16 parallels claim 1 and thus the outstanding rejection is defective as against claim 16 as well.

The rejection of independent claim 10 is also defective for similar reasons in that the claim recites, "generating an identifier for said atomic transaction for said user program; providing said identifier to said user program; receiving a plurality of combinations for execution from said user program, wherein each of said plurality of combinations contains said transaction identifier, …".

Claim 1 further recites, "specifying in said user program a plurality of combinations, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein each combination indicates that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted, …".

The Examiner first asserts:

> … The term "if denotes an optionally recited limitation and optionally recited limitations are not guaranteed to take place and are therefore not required to be taught, see MPEP § 2106 Section 11(C). (see Page 2 last but one paragraph of the Outstanding Final Office Action)

The Examiner clearly misinterprets the above quoted section of MPEP, which further states:

> The following are examples of language that may raise a question as to the limiting effect of the language in a claim:
> (A) statements of intended use or field of use,
> (B) "adapted to" or "adapted for" clauses,
> (C) "wherein" clauses, or
> (D) "whereby" clauses.
> This list of examples is not intended to be exhaustive. See also MPEP § 2111.04.

The Honorable Board's attention is directed to the fact that 'if' is not among the Examples (albeit is stated that they are not intended to be exhaustive). Furthermore, as stated in MPEP § 2111.04:

> The determination of whether each of these clauses is a limitation in a claim depends on the specific facts of the case. In *Hoffer v. Microsoft Corp.*, 405 F.3d 1326, 1329, 74 USPQ2d 1481, 1483 (Fed. Cir. 2005), the court held that when a "'whereby' clause states a condition that is material to patentability, it cannot be ignored in order to change the substance of the invention." *Id.*

Here, the two 'if' recitations denote the occurrence of corresponding claimed events (i.e., first that execution of the corresponding task procedure is completed and second that the atomic transaction is to be aborted). The occurrence of such recited events is a condition 'material to patentability' and thus cannot be ignored, per MPEP § 2111.04.

Applicants thank the Examiner for having continued Examination of the if recitations noted above, but assert that the comparison advanced by the Examiner is plain unreasonable. That is, one skilled in the relevant arts, in possession of the Applicant's disclosure, would not make such an analogy.

Specifically, the Examiner equates the claimed specifying noted above to the tuple Account1(), P2(), and do-reverse-of-P2() of Figure 1 of the subject patent application. Applicants again assert that such an analogy to be unreasonable. In particular, it is noted that there is **no inherent logic to the structure of Figure 1, which makes the tuple akin to the claimed combination**. The Examiner simply uses the Applicant's disclosure in finding such an analogy.

The rejection of all the independent claims is defective for this additional reason as well.

Claim 16 recites a computer system performing the actions including, “keep track of a set of rollback procedures corresponding to said set of task procedures,...”. The Examiner equates such keeping track to the programmer efforts in developing the program code of Figure 1. See page 4 second paragraph of the Outstanding Final Office Action.

It is respectfully pointed out that the specific operation of a computer system cannot reasonably be equated to the mental effort put-in by a programmer (the very overhead sought to be avoided by the present invention). The rejection of claim 16 (as well as 10) is defective for this reason as well.

VII.A.2. The References Would Not Work Together as Claimed, Even if Combined

Even assuming arguendo that the program of Figure 1 of the subject patent application is executed in the environment of Gostanian, it is noted that there is no disclosure or suggestion in Gostanian to provide the transaction identifier back to the user programs. The Honorable Board’s attention is drawn to the fact that Figures 4 and 5 of Gostanian merely show assignment of identification to transactions received from application client (AC) and then only returns the result of processing of transactions to AC (see bottom left of Figure 4 of Gostanian and Col. 13 lines 21-37 of Gostanian).

Thus, there is no disclosure or suggestion in Gostanian that the transaction identifier is provided to the application clients.

Lordi also does not cure such a deficiency in that file system of Lordi is believed to interface with a requesting system only after completion of performance of a requested operation (see, for example, Figures 3 and 4 of Lordi, where communication is believed to be only after ‘DONE’ state).

It is accordingly concluded that the references individually do not teach or suggest at least one claimed feature, and in addition even the combined operation of the references does not teach or suggest one or more claimed features.

VII.A.3. The Benefits of the Present Invention are NOT taught by any of the references individually

The Honorable Board's attention is directed to the fact that none of the three references individually provides the benefits of the present invention.

In particular, as noted in the 'Summary of the claimed subject matter' section above, (A) a programmer is relieved of the challenge of keeping track of the sequence of task procedures executed (based on execution flow), and (B) yet be able to provide the custom roll back procedures in the user program.

APAA does not provide benefit A noted above, as should be apparent from the Summary section of above.

Gostanian does not provide benefit B noted above (in user programs). As a support for this assertion, the Honorable Board's attention is directed to the fact that the techniques of Figures 4 and 5 of Gostanian, relied upon by the Examiner, are operative in server systems 332/334 (of Figure 3 of Gostanian), and there is no disclosure or suggestion that the rollback intelligence (e.g., commit/abort (rollback) 450/454/568/560 of Figures 4 and 5 of Gostanian) is provided by the user programs there.

Lordi does not provide benefit B above. As a support for this assertion, the Honorable Board's attention is drawn to Figures 5-14, wherein the 'undo' operation is pre-defined for each of the tasks there. For example, the specific undo operation for 'remove directory' task of Figure 5 of Lordi is pre-defined (steps 315 and 316), and thus feature B of above is not taught or reasonably suggested by Lordi.

From the above, it is asserted that the above observation is strong indicia that the Examiner is basing the conclusion of obviousness, based on nothing more than hindsight gleaned from Applicant's disclosure. MPEP § 707.07 (f) expressly requires consideration of the benefit while determining patentability over the art of record.

VII.A.4. Affidavits Establish a Long-felt Need the Present Invention Meets

Applicants had submitted an affidavit along with amendment dated 11 December 2009. The affidavit has shown a persistent need for several years for techniques to address support for the atomic transactions and absence of technique similar to that of the claimed invention. It is Applicants position that the absence of the claimed technique in any single reference, while there being a persistent need for simplication and/or more flexibility establishes a long felt need that is solved by the present invention.

The Examiner states that "Thus, any proof of long felt need must show a long felt need for the distinct embodiments of claim 1 and 7." It is respectfully noted that the combination of features recited in each of the independent claims has not been recognized in the prior art, and that makes it unobvious. When the combination is itself is not recognized, it is illogical to expect to find a need for the specific embodiment, short of a general need of having the combination of simplicity and flexibility provided by the present invention.

It is accordingly submitted that each of the independent claims has been established to be unobvious over the art of record.

B. Rejection of Claims 3-4, 14-15, 18 and 19 Under 35 U.S.C. § 103(a)

Claims 3-4, 14-15, 18 and 19 are allowable over the art of record at least for the reasons noted with respect to corresponding allowable base/independent claims.

The Honorable Board is accordingly respectfully requested to reverse all outstanding rejections under 35 U.S.C. § 103.

Conclusion

The Office is invited to at 443-552-7281 (4AM-noon EST, else voicemail) if it is believed that an interview might be useful for any reason. The Director is hereby authorized to charge any underpayment of fees (including extension fees), or credit any overpayments to Deposit Account No.: 20-0674.

Respectfully submitted,
/Narendra Reddy Thappeta/
Signature

Date: 21-Sep-10

Printed Name: Narendra Reddy Thappeta
Attorney for Appellant
Registration Number: 41,416

VIII. CLAIMS

Claim 1 (Previously Presented): A method of implementing atomic transactions in a system, said method comprising:

requesting in a user program a transaction identifier for an atomic transaction;

generating said transaction identifier in a transaction manager in response to said requesting;

specifying in said user program a plurality of combinations, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure,

wherein each combination indicates that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted,

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure;

executing a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations;

keeping track of a set of rollback procedures corresponding to said set of task procedures, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program, wherein said set of rollback procedures are kept track of external to said user program in response to said executing of the corresponding task procedures; and

executing said set of rollback procedures in a reverse order of said sequential order if said atomic transaction is to be aborted,

wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program,

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure, and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures.

Claim 2 (Original): The method of claim 1, wherein said transaction identifier is unique to each of the atomic transactions.

Claim 3 (Original): The method of claim 1, wherein said keeping comprises storing data representing said rollback procedures in a stack.

Claim 4 (Original): The method of claim 3, wherein said stack is stored in a memory.

Claim 5 (Previously Presented): The method of claim 1, wherein said user program further comprises additional instruction for examining a status returned by execution of one of said task procedures and performing said aborting if said status indicates an error,

wherein said aborting is specified in said user program using an instruction containing said transaction identifier.

Claim 6 (Original): The method of claim 1, wherein said aborting is performed asynchronously.

Claim 7 (Previously Presented): A computer readable storage medium carrying one or more sequences of instructions representing a user program for execution on a system, said user program implementing an atomic transaction, wherein execution of said one or more sequences of instructions by one or more processors contained in said system causes said system to perform the actions of:

requesting an identifier in said user program from a transaction manager for said atomic transaction, said transaction manager being provided external to said user program, wherein said transaction manager generates a unique value as said identifier and provides said identifier to said user program;

setting a variable to equal said identifier in said user program;

specifying a plurality of combinations in said user program for execution in said system, wherein each of said plurality of combinations contains said variable, a task procedure, and a rollback procedure,

wherein each combination indicates in said user program that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said variable in each of said plurality of combinations specifies said identifier generated by said transaction manager;

executing a set of task procedures in a sequential order, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations; and

aborting said atomic transaction by specifying, in said user program, said identifier associated with an abort procedure to cause a set of rollback procedures to be executed in a reverse order of said sequential order, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures,

wherein said plurality of combinations and said abort procedure are contained in said user program,

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure, and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures.

Claim 8 (Previously Presented): The computer readable storage medium of claim 7, wherein said specifying comprises including each of said plurality of combinations in a single procedure call.

Claim 9 (Previously Presented): The computer readable storage medium of claim 7, further comprising examining a status returned by execution of one of said task procedures and performing said aborting if said status indicates an error.

Claim 10 (Previously Presented): A computer readable storage medium carrying one or more sequences of instructions for supporting implementation of a transaction manager which supports an atomic transaction in a user program executing in a system, wherein execution of said one or more sequences of instructions by one or more processors contained in said system causes said system to perform the actions of:

generating an identifier for said atomic transaction for said user program;

providing said identifier to said user program;

receiving a plurality of combinations for execution from said user program, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure,

wherein each combination indicates that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted,

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure;

executing a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations;

keeping track of a set of rollback procedures corresponding to said set of task procedures, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program; and

executing said set of rollback procedures in a reverse order of said sequential order in response to receiving an abort request, said abort request being received from said user program and containing said identifier,

wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program,

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure, and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures,

wherein said transaction manager is provided external to user programs including said user program.

Claims 11 - 12 (Canceled)

Claim 13 (Previously Presented): The computer readable storage medium of claim 10, wherein said transaction identifier is generated to be unique for each atomic transaction.

Claim 14 (Previously Presented): The computer readable storage medium of claim 10, wherein said set of rollback procedures are represented in the form of a stack.

Claim 15 (Previously Presented): The computer readable storage medium of claim 14, wherein said stack is stored in a memory.

Claim 16 (Previously Presented): A computer system comprising:

a memory storing a plurality of instructions; and

a processing unit coupled to said memory and executing said plurality of instructions;

a computer readable medium to store and provide said plurality of instructions to said memory, wherein execution of said plurality of instructions by said processing unit causes said computer system to support implementation of atomic transactions in a programming environment by performing the actions of:

request in a user program, a transaction identifier for an atomic transaction;

generate said transaction identifier in a transaction manager in response to said requesting, wherein said transaction manager is provided external to said user program;

specify in said user program a plurality of combinations wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure,

wherein each combination indicates in said user program that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted,

wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure, wherein said rollback procedure is specified as a separate procedure from said task procedure;

execute a set of task procedures in a sequential order according to said user program, wherein said set of task procedures are contained in said task procedures specified in said plurality of combinations;

keep track of a set of rollback procedures corresponding to said set of task procedures, each of said set of rollback procedures being determined based on a combination corresponding to an executed task procedure contained in said set of task procedures, said combination being contained in said plurality of combinations specified in said user program, wherein said set of rollback procedures are kept track of external to said user program in response to said executing of the corresponding task procedures; and execute said set of rollback procedures in a reverse order of said sequential order if said atomic transaction is to be aborted, wherein said rollback procedures are identified according to said keeping,

wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure, and

whereby each user program has corresponding custom logic specified by a user for each of the rollback procedures.

Claim 17 (Original): The computer system of claim 16, wherein said transaction identifier is unique to each of the atomic transactions.

Claim 18 (Previously Presented): The computer system of claim 16, wherein the actions performed by said computer system further comprise store data representing said rollback procedures in a stack to perform said keep.

Claim 19 (Original): The computer system of claim 18, wherein said stack is stored in a memory.

Claim 20 (Previously Presented): The computer system of claim 16, wherein the actions performed by said computer system further comprise examine a status returned by execution of one of said task procedures and to perform said aborting if said status indicates an error.

Claim 21 (Previously Presented): The computer system of claim 16, wherein the actions performed by said computer system further comprise execute said rollback procedures asynchronously.

Claims 22 - 24 (Canceled)

Claim 25 (Previously Presented): The computer readable storage medium of claim 7, wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None